PBxplore: a tool to analyze local protein structure and deformability with Protein Blocks

Jonathan Barnoud[1,2,3,4,6,*], Hubert Santuz[1,2,3,4,7,*], Pierrick Craveur[1,2,3,4,8], Agnel Praveen Joseph[1,2,3,4,9], Vincent Jallu[5], Alexandre G. de Brevern[1,2,3,4,*] and Pierre Poulain[1,2,3,4,10,*]

[1] INSERM, U 1134, DSIMB, Paris, France
[2] Univ. Paris Diderot, Sorbonne Paris Cité, Univ de la Réunion, Univ des Antilles, UMR-S 1134, Paris, France
[3] Institut National de la Transfusion Sanguine (INTS), Paris, France
[4] Laboratoire d'Excellence GR-Ex, Paris, France
[5] Platelet Unit, INTS, Paris, France
[6] Current affiliation: Groningen Biomolecular Sciences and Biotechnology Institute and Zernike Institute for Advanced Materials, University of Groningen, Groningen, The Netherlands
[7] Current affiliation: Laboratoire de Biochimie Théorique, CNRS UPR 9080, Institut de Biologie Physico-Chimique, Paris, France
[8] Current affiliation: Department of Integrative Structural and Computational Biology, The Scripps Research Institute, La Jolla, CA, United States of America
[9] Current affiliation: Birkbeck College, University of London, London, UK
[10] Current affiliation: Mitochondria, Metals and Oxidative Stress Group, Institut Jacques Monod, UMR 7592, Univ. Paris Diderot, CNRS, Sorbonne Paris Cité, Paris, France
[*] These authors contributed equally to this work.

Submitted 28 August 2017
Accepted 19 October 2017
Published 20 November 2017

Corresponding authors
Alexandre G. de Brevern, alexandre.debrevern@univ-paris-diderot.fr
Pierre Poulain, pierre.poulain@univ-paris-diderot.fr

Academic editor
Walter de Azevedo Jr

Additional Information and Declarations can be found on page 12

DOI **10.7717/peerj.4013**

ABSTRACT

This paper describes the development and application of a suite of tools, called PBxplore, to analyze the dynamics and deformability of protein structures using Protein Blocks (PBs). Proteins are highly dynamic macromolecules, and a classical way to analyze their inherent flexibility is to perform molecular dynamics simulations. The advantage of using small structural prototypes such as PBs is to give a good approximation of the local structure of the protein backbone. More importantly, by reducing the conformational complexity of protein structures, PBs allow analysis of local protein deformability which cannot be done with other methods and had been used efficiently in different applications. PBxplore is able to process large amounts of data such as those produced by molecular dynamics simulations. It produces frequencies, entropy and information logo outputs as text and graphics. PBxplore is available at https://github.com/pierrepo/PBxplore and is released under the open-source MIT license.

Subjects Bioinformatics, Computational Biology, Molecular Biology
Keywords Protein blocks, Deformability, Python, Protein, Structure, Structural alphabet

INTRODUCTION

Proteins are highly dynamic macromolecules (*Frauenfelder, Sligar & Wolynes, 1991*; *Bu & Callaway, 2011*). To analyze their inherent flexibility, computational biologists often use molecular dynamics (MD) simulations. The quantification of protein flexibility is based

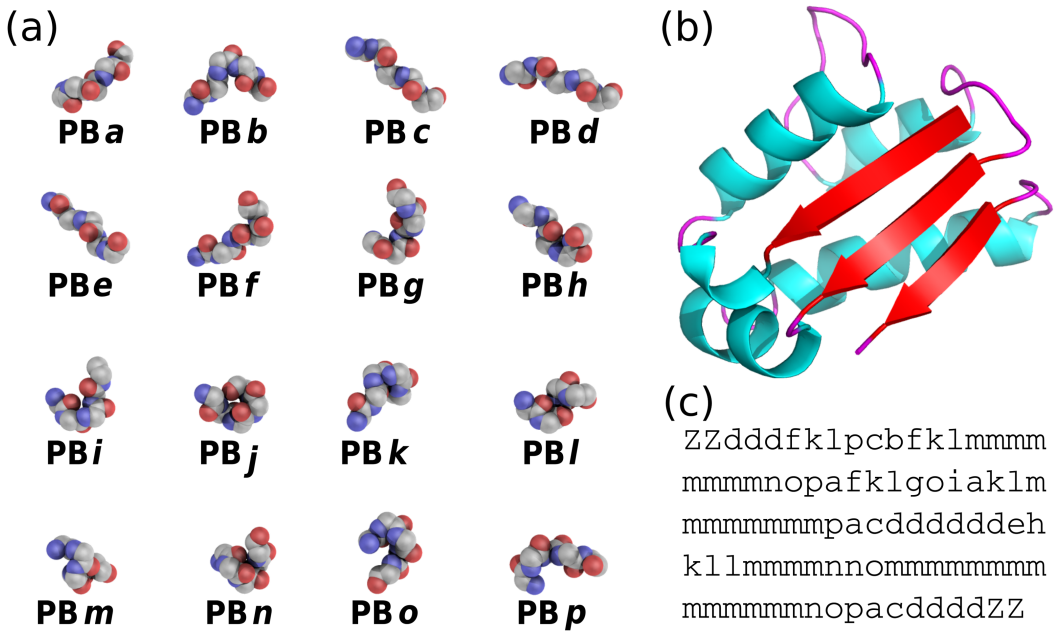

Figure 1 (A) The 16 protein blocks (PBs) represented in balls with carbon atoms in gray, oxygen atoms in red and nitrogen atoms in purple (hydrogen atoms are not represented). (B) The barstar protein (PDB ID 1AY7 (*Sevcík et al., 1998*)) represented in cartoon with alpha-helices in blue, beta-strands in red and coil in pink. These representations were generated using PyMOL software (*DeLano, 2002*) (C) PBs sequence obtained from PBs assignment. Z is a dummy PB, meaning that no PB can be assigned to this position.

Full-size DOI: 10.7717/peerj.4013/fig-1

on various methods such as Root Mean Square Fluctuations (RMSF) that rely on multiple MD snapshots or Normal Mode Analysis (NMA) that rely on a single structure and focus on quantifying large movements.

Alternative *in silico* approaches assess protein motions through the protein residue network (*Atilgan, Turgut & Atilgan, 2007*) or dynamical correlations from MD simulations (*Ghosh & Vishveshwara, 2007*; *Dixit & Verkhivker, 2011*). Another noticeable development is the MOdular NETwork Analysis (MONETA), which localizes the perturbations propagation throughout a protein structure (*Laine, Auclair & Tchertanov, 2012*).

Here we use an alternative yet powerful approach based on small prototypes or "structural alphabets" (SAs). SAs approximate conformations of protein backbones and code the local structures of proteins as one-dimensional sequences (*Offmann, Tyagi & De Brevern, 2007*). Protein Blocks (PBs) (*De Brevern, Etchebest & Hazout, 2000*) are one of these SAs (*De Brevern, 2005*; *Etchebest et al., 2005*; *Joseph et al., 2010*).

PBs are composed of 16 blocks designed through an unsupervised training performed on a representative non-redundant databank of protein structures (*De Brevern, Etchebest & Hazout, 2000*). PBs are defined from a set of dihedral angles describing the protein backbone. This property makes PBs interesting conformational prototypes of the local protein structure. PBs are labeled from *a* to *p* (see Fig. 1A). PBs *m* and *d* are prototypes for central α-helix and central β-strand, respectively. PBs *a* to *c* primarily represent β-strand

N-caps and PBs *e* and *f*, β-strand C-caps; PBs *g* to *j* are specific to coils, PBs *k* and *l* are specific to α-helix N-caps, and PBs *n* to *p* to α-helix C-caps (*De Brevern, 2005*). Figure 1 illustrates how a PB sequence is assigned from a protein structure. Starting from the 3D coordinates of the barstar protein (Fig. 1B), the local structure of each amino acid is compared to the 16 PB definitions (Fig. 1A). The most similar protein block is assigned to the residue under consideration (the similarity metric is explained latter in this article). Eventually, assignment leads to the PB sequence represented in Fig. 1C.

By reducing the complexity of protein structure, PBs have been shown to be efficient and relevant in a wide spectrum of applications. To name a few, PBs have been used to analyze protein contacts (*Faure, Bornot & De Brevern, 2008*), to propose a structural model of a transmembrane protein (*De Brevern, 2005*), to reconstruct globular protein structures (*Dong, Wang & Lin, 2007*), to design peptides (*Thomas et al., 2006*), to define binding site signatures (*Dudev & Lim, 2007*), to perform local protein conformation predictions (*Li, Zhou & Liu, 2009*; *Rangwala, Kauffman & Karypis, 2009*; *Suresh, Ganesan & Parthasarathy, 2013*; *Suresh & Parthasarathy, 2014*; *Zimmermann & Hansmann, 2008*), to predict β-turns (*Nguyen et al., 2014*) and to understand local conformational changes due to mutations of the αIIb β3 human integrin (*Jallu et al., 2012*; *Jallu et al., 2013*; *Jallu et al., 2014*).

PBs are also useful to compare and superimpose protein structures with pairwise and multiple approaches (*Joseph, Srinivasan & De Brevern, 2011*; *Joseph, Srinivasan & De Brevern, 2012*), namely iPBA (*Gelly et al., 2011*) and mulPBA (*Léonard et al., 2014*), both currently showing the best results compared to other superimposition methods. Eventually, PBs lead to interesting results at predicting protein structures from their sequences (*Ghouzam et al., 2015*; *Ghouzam et al., 2016*) and at predicting protein flexibility (*Bornot, Etchebest & de Brevern, 2011*; *De Brevern et al., 2012*).

Applying PB-based approaches to biological systems such as the DARC protein (*De Brevern et al., 2005*), the human αIIb β3 integrin (*Jallu et al., 2012*; *Jallu et al., 2013*; *Jallu et al., 2014*) and the KISSR1 protein (*Chevrier et al., 2013*) highlighted the usefulness of PBs in understanding local deformations of large protein structures. Specifically, these analyzes have shown that a region considered as highly flexible through RMSF quantifications can be seen using PBs as locally highly rigid. This unexpected behavior is explained by a local rigidity surrounded by deformable regions (*Craveur et al., 2015*). To go further, we recently used PBs to analyze long-range allosteric interactions in the Calf-1 domain of αIIb integrin (*Goguet et al., 2017*). To our knowledge, the only other related approach based on SA to assess local deformation is GSATools (*Pandini et al., 2013*); it is specialized in the analysis of functional correlations between local and global motions, and the mechanisms of allosteric communication.

Despite the versatility of PBs and the large spectrum of their applications, PBs lack a uniform and easy-to-use toolkit to assign PB sequences from 3D structures, and to analyze these sequences. The only known implementation is a an old C program not publicly available and not maintained anymore. Such a tool not being available limits the usability of the PBs for studies where they would be meaningful.

We thus propose PBxplore, a tool to analyze local protein structure and deformability using PBs. It is available at https://github.com/pierrepo/PBxplore. PBxplore

can read PDB structure files (*Bernstein et al., 1977*), PDBx/mmCIF structure files (*Bourne et al., 1997*), and MD trajectory formats from most MD engines, including Gromacs MD topology and trajectory files (*Lindahl, Hess & Van der Spoel, 2001*; *Van der Spoel et al., 2005*). Starting from 3D protein structures, PBxplore assigns PBs sequences; it computes a local measurement of entropy, a density map of PBs along the protein sequence and a WebLogo-like representation of PBs.

In this paper, we first present the principle of PBxplore, then its different tools, and finally a step-by-step user-case with the $\beta 3$ subunit of the human platelet integrin αIIb $\beta 3$.

DESIGN AND IMPLEMENTATION

PBxplore is written in Python (*Van Rossum, 1995*; *Python Software Foundation, 2010*; *Bassi, 2007*). It is compatible with Python 2.7, and with Python 3.4 or greater. It requires the Numpy Python library for array manipulation (*Ascher et al., 1999*), the matplotlib library for graphical representations, and the MDAnalysis library for molecular dynamics simulation files input (*Michaud-Agrawal et al., 2011*; *Gowers et al., 2016*). Optionally, PBxplore functionalities can be enhanced by the installation and the use of WebLogo (*Crooks et al., 2004*) to create sequence logos.

PBxplore is available as a set of command-line tools and as a Python module. The command-line tools allow easy integration of PBxplore in existing analysis pipelines. These programs can be linked up together to carry out the most common analyses on PB sequences to provide insights on protein flexibility. In addition, the PBxplore Python library provides an API to access its core functionalities which allows the integration of PBxplore in Python programs and workflows, and the extension of the method to suit new needs.

PBxplore is released under the open-source MIT license (https://opensource.org/licenses/MIT). It is available on the software development platform GitHub at https://github.com/pierrepo/PBxplore.

The package contains unit and regression tests and is continuously tested using Travis CI (https:// travis-ci.org/). An extensive documentation is available on Read the Docs (*Holscher, Leifer & Grace, 2010*) at https://pbxplore.readthedocs.io.

Installation

The easiest way to install PBxplore is through the Python Package Index (PyPI):

```
pip install --user pbxplore
```

It will ensure all required dependencies are installed correctly.

Command-line tools

A schematic description of PBxplore command line interface is provided in Fig. 2. The interface is composed of three different programs: `PBassign` to assign PBs, `PBcount` to compute PBs frequency on multiple conformations, and `PBstat` to perform statistical analyses and visualization. These programs can be linked up together to make a structure analysis pipeline to study protein flexibility.

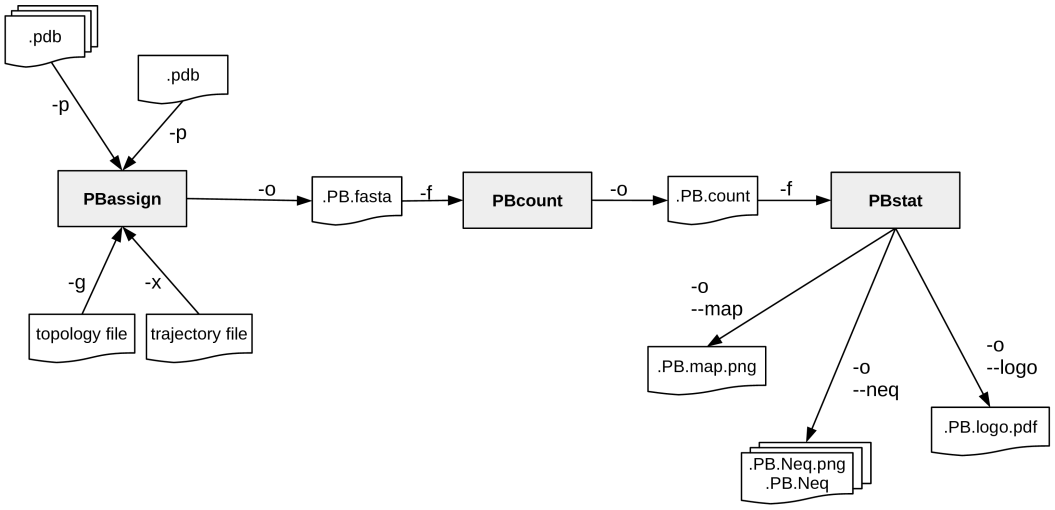

Figure 2 PBxplore is based on 3 programs that can be chained to build a structure analysis pipeline. Main input file types (.pdb, MD trajectory, MD topology), output files (.fasta, .png, .Neq, .pdf) and parameters (beginning with a single or double dash) are indicated.

Full-size DOI: 10.7717/peerj.4013/fig-2

PBassign

The very first task is to assign PBs from the protein structure(s). A PB is associated to each pentapeptide in the protein chain. To assign a PB to the residue n, five residues are required (residues $n-2$, $n-1$, n, $n+1$ and $n+2$). From the backbone conformation of these five residues, eight dihedral angles (ψ and ϕ) are computed, going from the ψ angle of residue $n-2$ to the ϕ angle of residue $n+2$ (*De Brevern, 2005*). This set of eight dihedral angles is then compared to the reference angles set for the 16 PBs (*De Brevern, Etchebest & Hazout, 2000*) using the Root Mean Square Deviation Angle (RMSDA) measure, i.e., an Euclidean distance on angles. PB with the smallest RMSDA is assigned to residue n. A dummy PB Z is assigned to residues for which all eight angles cannot be computed. Hence, the first two N-terminal and the last two C-terminal residues are always assigned to PB Z.

The program `PBassign` reads one or several protein 3D structures and performs PBs assignment as one PBs sequence per input structure. `PBassign` can process multiple structures at once, either provided as individual structure files or as a directory containing many structure files or as topology and trajectory files obtained from MD simulations. Note that PBxplore is able to read any trajectory file format handled by the MDAnalysis library, yet our tests focused on Gromacs trajectories. Output PBs sequences are bundled in a single file in FASTA format.

PBcount

During the course of a MD simulation, the local protein conformations can change. It is then interesting to analyze them through PB description. Indeed, as each PB describes a local conformation, the variability of the PB assigned to a given residue throughout the trajectory indicates some local deformation of the protein structure. Thus, once PBs are assigned, PBs frequencies per residue can be computed.

The program `PBcount` reads PBs sequences for different conformations of the same protein from a file in FASTA format (as outputted by `PBassign`). Many input files can be provided at once. The output data is a 2D matrix of x rows by y columns, where x is the length of the protein sequence and y is the 16 distinct PBs. A matrix element is the count of a given PB at a given position in the protein sequence.

PBstat

The number of possible conformational states covered by PBs is higher than the classical secondary structure description (16 states instead of 3). As a consequence, the amount of information produced by `PBcount` can be complex to handle. Hence, we propose three simple ways to visualize the variation of PBs which occur during a MD simulation.

The program `PBstat` reads PBs frequencies as computed by `PBcount`. It can produce three types of outputs based on the input argument(s). The first two use the matplotlib library and the last one requires the installation of the third-party tool Weblogo (*Crooks et al., 2004*). `PBstat` also offers two options (`--residue-min` and `--residue-max`) to define a residue frame allowing the user to quickly look at segments of interest. The three graphical representations proposed are:

- *Distribution of PBs.* This feature plots the frequency of each PB along the protein sequence. The output file could be in format .png, .jpg or .pdf. A dedicated colorblind safe color range (*Brewer et al., 2013*) allows visualizing the distribution of PBs. For a given position in the protein sequence, blue corresponds to a null frequency when the particular PB is never sampled at this position and red corresponds to a frequency of 1 when the particular PB is always found at this position. This representation is produced with the `--map` argument.
- *Equivalent number of PBs (N_{eq}).* The N_{eq} is a statistical measurement similar to entropy (*Offmann, Tyagi & De Brevern, 2007*). It represents the average number of PBs sampled by a given residue. N_{eq} is calculated as follows:

$$N_{eq} = \exp\left(-\sum_{i=1}^{16} f_x \ln f_x\right)$$

  where f_x is the probability (or frequency) of the PB x. A N_{eq} value of 1 indicates that only a single type of PB is observed, while a value of 16 is equivalent to a random distribution, i.e., all PBs are observed with the same frequency 1/16. For example, a N_{eq} value around 5 means that, across all the PBs observed at the position of interest, 5 different PBs are mainly observed. If the N_{eq} exactly equals to 5, this means that 5 different PBs are observed in equal proportions (i.e., 1/5).

  A high N_{eq} value can be associated with a local deformability of the structure whereas a N_{eq} value close to 1 means a rigid structure. In the context of structures issued from MD simulations, the concept of deformability / rigidity is independent to the one of mobility. The N_{eq} representation is produced with the `--neq` argument.
- *Logo representation of PBs frequency.* This is a WebLogo-like representation (*Crooks et al., 2004*) of PBs sequences. The size of each PB is proportional to its frequency at a given

position in the sequence. This type of representation is useful to pinpoint PBs patterns. This WebLogo-like representation is produced with the `--logo` argument.

Python module

PBxplore is also a Python module that more advanced users can embed in their own Python script. Here is a Python 3 example that assigns PBs from the structure of the barstar ribonuclease inhibitor (*Lubienski et al., 1994*):

```
import urllib.request
import pbxplore as pbx

# Download the pdb file
urllib.request.urlretrieve ('https://files.rcsb.org/view/1BTA.pdb',
'1BTA.pdb')

# The function pbx.chain_from_files () reads a list of files
# and for each one returns the chain and its name.
for chain_name, chain in pbx.chains_from_files (['1BTA.pdb']):
    # Compute phi and psi angles
    dihedrals = chain.get_phi_psi_angles ()
    # Assign PBss
    pb_seq = pbx.assign(dihedrals)
    print ('PBs sequence for chain {}:\n {}'.format (chain_name,
    pb_seq))
```

The documentation contains complete and executable Jupyter notebooks explaining how to use the module. It goes from the PBs assignments to the visualization of the protein deformability using the analysis functions. This allows the user to quickly understand the architecture of the module.

RESULTS

This section aims at giving the reader a quick tour of PBxplore features on a real-life example. We will focus on the $\beta 3$ subunit of the human platelet integrin αIIb$\beta 3$ that plays a central role in hemostasis and thrombosis. The $\beta 3$ subunit has also been reported in cases of alloimmune thrombocytopenia (*Kaplan, 2006*; *Kaplan & Freedman, 2007*). We studied this protein by MD simulations (for more details, see references (*Jallu et al., 2012*; *Jallu et al., 2013*; *Jallu et al., 2014*)).

The $\beta 3$ integrin subunit structure (*Poulain & De Brevern, 2012*) comes from the structure of the integrin complex (PDB 3FCS (*Zhu et al., 2008*)). Final structure has 690 residues and was used for MD simulations. All files mentioned below are available in the `demo_paper` directory from the GitHub repository (https://github.com/pierrepo/PBxplore/tree/master/demo_paper).

Protein blocks assignment

The initial file `beta3.pdb` contains 225 structures issued from a single 50 ns MD simulation of the $\beta 3$ integrin.

```
PBassign -p beta3.pdb -o beta3
```

This instruction generates the file `beta3.PB.fasta`. It contains as many PB sequences as there are structures in the input `beta3.pdb` file.

Protein Blocks assignment is the slowest step. In this example, it took roughly 80 s on a laptop with a quad-core-1.6-GHz processor.

Protein blocks frequency

```
PBcount -f beta3.PB.fasta -o beta3
```

The above command line produces the file `beta3.PB.count` that contains a 2D-matrix with 16 columns (as many as different PBs) and 690 rows (one per residue) plus one supplementary column for residue number and one supplementary row for PBs labels.

Statistical analysis

Distribution of PBs

```
PBstat -f beta3.PB.count -o beta3 --map
```

Figure 3 shows the distribution of PBs for the $\beta 3$ integrin. The color scale ranges from blue (the PB is not found at this position) to red (the PB is always found at this position). The $\beta 3$ protein counts 690 residues. This leads to a cluttered figure and prevents getting any details on a specific residue (Fig. 3A). However, it exhibits some interesting patterns colored in red that correspond to series of neighboring residues exhibiting a fixed PB during the entire MD simulation. See for instance patterns associated to PBs *d* and *m* that reveal β-sheets and α-helices secondary structures (*De Brevern, 2005*).

With a large protein such as this one, it is better to look at limited segments. A focus on the PSI domain (residue 1 to 56) (*Jallu et al., 2012*; *Zhu et al., 2008*) of the $\beta 3$ integrin was achieved with the command:

```
PBstat -f beta3.PB.count -o beta3 --map --residue-min 1 --residue-max 56
```

Figure 3B shows the PSI domain dynamics in terms of PBs. Interestingly, residue 33 is the site of the human platelet antigen (HPA)-1 alloimmune system. It is the first cause of alloimmune thrombocytopenia in Caucasian populations and a risk factor for thrombosis (*Kaplan, 2006*; *Kaplan & Freedman, 2007*). In Fig. 3B, this residue occupies a stable conformation with PB *h*. Residues 33 to 35 define a stable core composed of PBs *h-i-a*. This core is found in all of the 255 conformations extracted from the MD simulation and then is considered as highly rigid. On the opposite, residue 52 is flexible as it is found associated to PBs *i*, *j*, *k* and *l* corresponding to coil and α-helix conformations.

Equivalent number of PBs

The N_{eq} is a statistical measurement similar to entropy and is related to the flexibility of a given residue. The higher is the value, the more flexible is the backbone. The N_{eq} for the PSI domain (residue 1 to 56) was obtained from the command line:

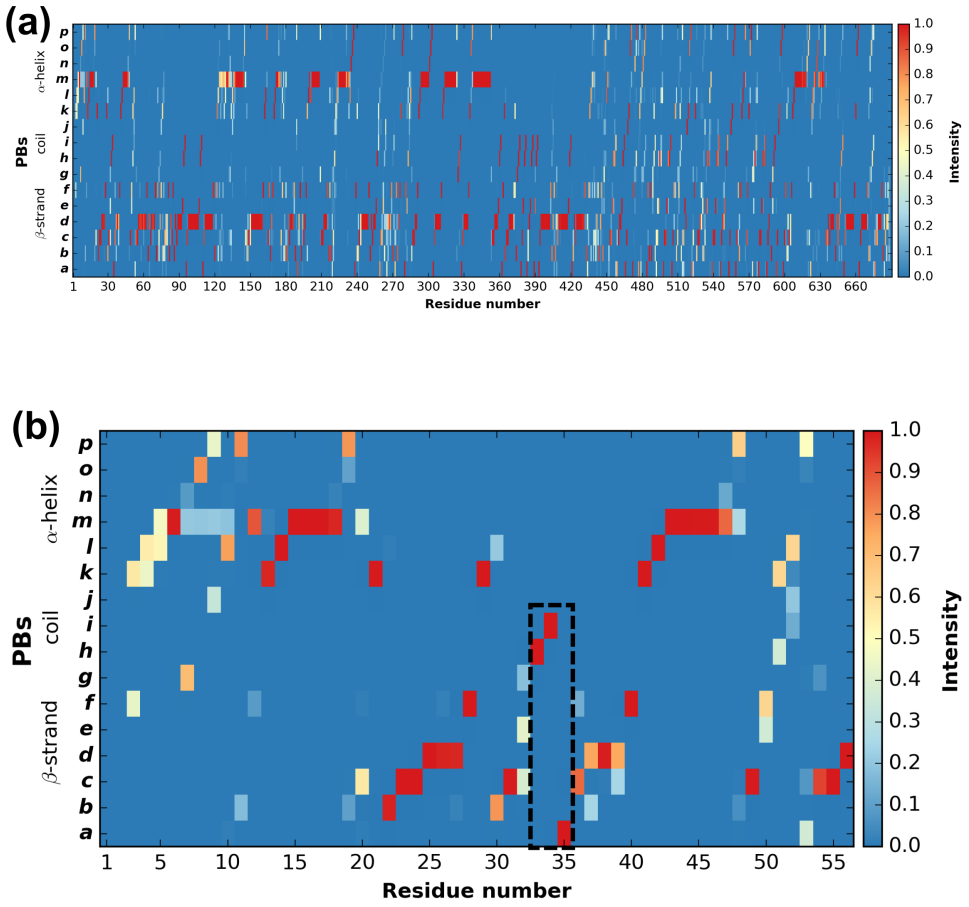

Figure 3 Distribution of PBs for the β3 integrin along the protein sequence. On the x-axis are found the 690 position residues and on the y-axis the 16 consecutive PBs from a to p (the two first and two last positions associated to "Z" have no assignment): (A) for the entire protein; (B) for the PSI domain only (residues 1 to 56). The dashed zone pinpoints residue 33 to 35.

Full-size DOI: 10.7717/peerj.4013/fig-3

```
PBstat -f beta3.PB.count -o beta3 --neq --residue-min 1 --residue-max 56
```

The output file `beta3.PB.Neq.1-56` contains two columns, corresponding to the residue numbers and the N_{eq} values. Figure 4A represents the N_{eq} along with the PBs sequence of the PSI domain, as generated by `PBstat`. The rigid region 33–35 and the flexible residue 52 are easily spotted, with low N_{eq} values for the former and a high N_{eq} value for the latter.

An interesting point, seen in our previous studies, is that the region delimited by residues 33 to 35 was shown to be highly mobile by the RMSF analysis we performed in *Jallu et al. (2012)*. RMSF was calculated on C-α atoms on the whole protein, for more details, see 'Materials and Methods' section in *Jallu et al. (2012)*. For comparison, RMSF and N_{eq} are represented on the same graph on Fig. 4B. This high mobility was correlated with the location of this region in a loop, which globally moved a lot in our MD simulations. Here, we observe that the region 33–35 is rigid. The high values of RMSF we observed in our previous work were due to flexible residues in the vicinity of the region 33–35,

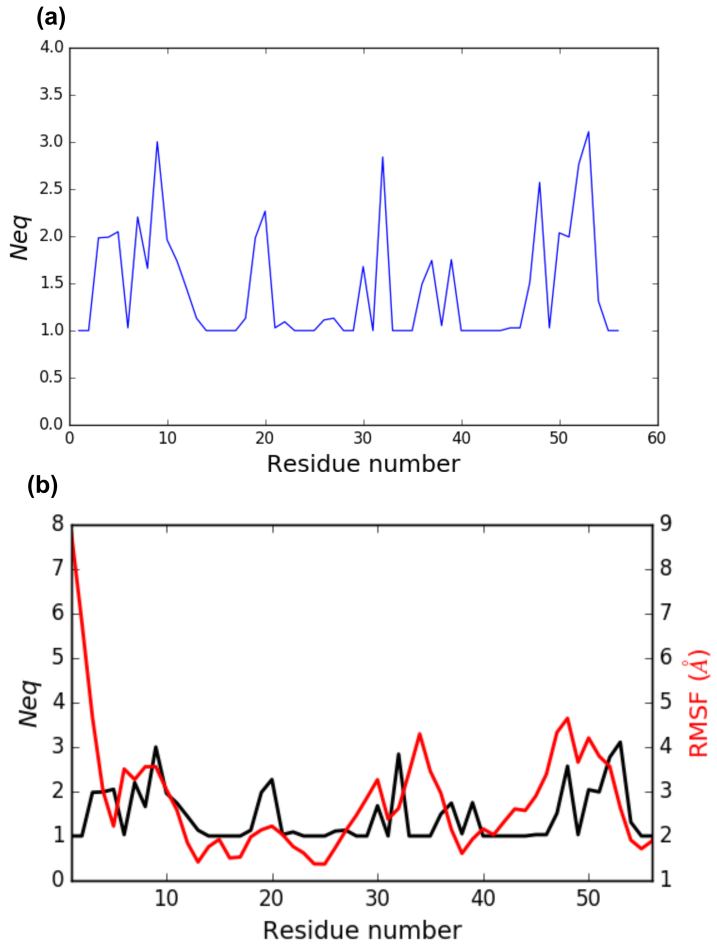

Figure 4 **(A) N_{eq} versus residue number for the PSI domain (residues 1 to 56); (B) comparison between RMSF and N_{eq}.**

Full-size DOI: 10.7717/peerj.4013/fig-4

probably acting as hinges (residues 32 and 36–37). Those hinges, due to their flexibility, induced the mobility of the whole loop: the region 33–35 fluctuated but did not deform. Understanding the flexibility of residues 33 to 35 is important since this region defines the HPA-1 alloantigenic system involved in severe cases of alloimmune thrombocytopenia. PBxplore allows discriminating between flexible and rigid residues. The N_{eq} is a metric of deformability and flexibility whereas RMSF quantifies mobility.

Logo representation of PBs frequency

While the N_{eq} analysis focuses on the flexibility of amino acids, the WebLogo-like representation (*Crooks et al., 2004*) aims at identifying the diversity of PBs and their frequencies at a given position in the protein sequence. With a focus on the PSI domain, the following command line was used:

```
PBstat -f beta3.PB.count -o beta3 --logo --residue-min 1 --residue-max 56
```

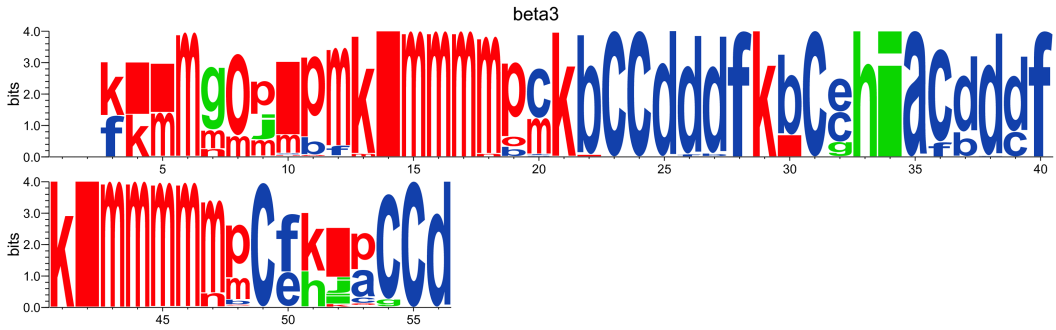

Figure 5 WebLogo-like representation of PBs for the PSI domain of the β3 integrin. PBs in red roughly correspond to α-helices, PBs in blue to β-sheets and PBs in green to coil.
Full-size DOI: 10.7717/peerj.4013/fig-5

Figure 5 represents PBs found at a given position. The rigid region 33–35 is composed of a succession of PBs *h-i-a* while the flexible residue 52 is associated to PBs *i*, *j*, *k* and *l*. This third representation summarized pertinent information, as shown in *Jallu et al. (2013)*.

CONCLUSION

From our previous works (*Jallu et al., 2012*; *Jallu et al., 2013*; *Jallu et al., 2014*; *Chevrier et al., 2013*), we have seen the usefulness of a tool dedicated to the analysis of local protein structures and deformability with PBs. We also showed the relevance of studying molecular deformability in the scope of structures issued from MD simulations. In a very recent study (*Goguet et al., 2017*), long independent MD simulations were performed for seven variants and one reference structure of the Calf-1 domain of the αIIb human integrin. Simulations were analyzed with PBxplore. Common and flexible regions as well as deformable zones were observed in all the structures. The highest B-factor region of Calf-1, usually considered as most flexible, is in fact a rather rigid region encompassed into two deformable zones. Each mutated structure barely showed any modifications at the mutation sites while distant conformational changes were detected by PBxplore. These results highlight the relevance of MD simulations in the study of both short and long range effects on protein structures, and demonstrate how PBs can bring insight from such simulations. In this context, we propose PBxplore, freely available at https://github.com/pierrepo/PBxplore. It is written in a modular fashion that allows embedding in any PBs related Python application.

SOFTWARE AVAILABILITY

PBxplore is released under the open-source MIT license (https://opensource.org/licenses/MIT). Its source code can be freely downloaded from the GitHub repository of the project: https://github.com/pierrepo/PBxplore. In addition, the present version of PBxplore (1.3.8) is also archived in the digital repository Zenodo (*Barnoud et al., 2017*).

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work was supported by grants from the National Institute for Blood Transfusion (INTS, France) and the Lab of Excellence GR-Ex to Jonathan Barnoud, Hubert Santuz, Pierrick Craveur, Agnel Praveen Joseph, Vincent Jallu, Alexandre G. de Brevern and Pierre Poulain; and from the Ministry of Research (France), University Paris Diderot, Sorbonne Paris Cité (France), National Institute for Health and Medical Research (INSERM, France) to Jonathan Barnoud, Hubert Santuz, Pierrick Craveur, Agnel Praveen Joseph, Alexandre G. de Brevern and Pierre Poulain. The labex GR-Ex, reference ANR-11-LABX-0051 is funded by the program ''Investissements d'avenir'' of the French National Research Agency, reference ANR-11-IDEX-0005-02. Alexandre G. de Brevern was also supported by the Indo-French Centre for the Promotion of Advanced Research/CEFIPRA grant (number 5302-2). Jonathan Barnoud was also supported by the TOP program of Prof. Marrink, financed by the Netherlands Organisation for Scientific Research (NWO). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
National Institute for Blood Transfusion.
Lab of Excellence GR-Ex: ANR-11-LABX-0051.
Ministry of Research.
University Paris Diderot.
Sorbonne Paris Cité.
National Institute for Health and Medical Research.
French National Research Agency: ANR-11-IDEX-0005-02.
Indo-French Centre for the Promotion of Advanced Research/CEFIPRA: 5302-2.
Netherlands Organisation for Scientific Research (NWO).

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Jonathan Barnoud and Hubert Santuz performed the experiments, analyzed the data, contributed reagents/materials/analysis tools, wrote the paper, prepared figures and/or tables, reviewed drafts of the paper.
- Pierrick Craveur, Agnel Praveen Joseph and Vincent Jallu contributed reagents/materials/analysis tools, reviewed drafts of the paper.
- Alexandre G. de Brevern conceived and designed the experiments, analyzed the data, contributed reagents/materials/analysis tools, wrote the paper, prepared figures and/or tables, reviewed drafts of the paper.

- Pierre Poulain conceived and designed the experiments, performed the experiments, analyzed the data, contributed reagents/materials/analysis tools, wrote the paper, prepared figures and/or tables, reviewed drafts of the paper.

Data Availability

The following information was supplied regarding data availability:

GitHub: Available at https://github.com/pierrepo/PBxplore

Zenodo: Available at https://dx.doi.org/10.5281/zenodo.1016257.

REFERENCES

Ascher D, Dubois PF, Hinsen K, James JH, Oliphant T. 1999. Numerical Python. Technical Report UCRL-MA-128569. Lawrence Livermore National Laboratory, Livermore, CA.

Atilgan AR, Turgut D, Atilgan C. 2007. Screened Nonbonded interactions in native proteins manipulate optimal paths for robust residue communication. *Biophysical Journal* **92(9)**:3052–3062 DOI 10.1529/biophysj.106.099440.

Barnoud J, Santuz H, De Brevern AG, Poulain P. 2017. PBxplore (v1.3.8): a program to explore protein structures with Protein Blocks. *Zenodo*.

Bassi S. 2007. A primer on python for life science researchers. *PLOS Computational Biology* **3(11)**:e199 DOI 10.1371/journal.pcbi.0030199.

Bernstein FC, Koetzle TF, Williams GJ, Meyer EF, Brice MD, Rodgers JR, Kennard O, Shimanouchi T, Tasumi M. 1977. The Protein Data Bank: a computer-based archival file for macromolecular structures. *Journal of Molecular Biology* **112(3)**:535–542.

Bornot A, Etchebest C, De Brevern AG. 2011. Predicting protein flexibility through the prediction of local structures. *Proteins* **79(3)**:839–852 DOI 10.1002/prot.22922.

Bourne PE, Berman HM, McMahon B, Watenpaugh KD, Westbrook JD, Fitzgerald PM. 1997. [30] Macromolecular crystallographic information file. In: *Methods in enzymology*, vol. 277. Amsterdam: Elsevier, 571–590.

Brewer C, Harrower M, Sheesley B, Woodruff A, Heyman D. 2013. *ColorBrewer2*. Hewitt: Axis Maps LLC.

Bu Z, Callaway DJ. 2011. Proteins MOVE! Protein dynamics and long-range allostery in cell signaling. In: *Advances in protein chemistry and structural biology*, vol. 83. Amsterdam: Elsevier, 163–221.

Chevrier L, De Brevern A, Hernandez E, Leprince J, Vaudry H, Guedj AM, De Roux N. 2013. PRR repeats in the intracellular domain of KISS1R are important for its export to cell membrane. *Molecular Endocrinology* **27(6)**:1004–1014 DOI 10.1210/me.2012-1386.

Craveur P, Joseph AP, Esque J, Narwani TJ, Noel F, Shinada N, Goguet M, Leonard S, Poulain P, Bertrand O, Faure G, Rebehmed J, Ghozlane A, Swapna LS, Bhaskara RM, Barnoud J, Téletchéa S, Jallu V, Cerny J, Schneider B, Etchebest C, Srinivasan N, Gelly J-C, De Brevern AG. 2015. Protein flexibility in the light of structural alphabets. *Frontiers in Molecular Biosciences* **2**:Article 20 DOI 10.3389/fmolb.2015.00020.

Crooks GE, Hon G, Chandonia J-M, Brenner SE. 2004. WebLogo: a sequence logo generator. *Genome Research* **14(6)**:1188–1190 DOI 10.1101/gr.849004.

De Brevern A, Wong H, Tournamille C, Colin Y, Le Van Kim C, Etchebest C. 2005. A structural model of a seven-transmembrane helix receptor: the Duffy Antigen/Receptor for Chemokine (DARC). *Biochimica et Biophysica Acta (BBA)–General Subjects* **1724(3)**:288–306 DOI 10.1016/j.bbagen.2005.05.016.

De Brevern AG. 2005. New assessment of a structural alphabet. *In Silico Biology* **5(3)**:283–289.

De Brevern AG, Bornot A, Craveur P, Etchebest C, Gelly J-C. 2012. PredyFlexy: flexibility and local structure prediction from sequence. *Nucleic Acids Research* **40(Web Server issue)**:W317–W322 DOI 10.1093/nar/gks482.

De Brevern AG, Etchebest C, Hazout S. 2000. Bayesian probabilistic approach for predicting backbone structures in terms of protein blocks. *Proteins* **41(3)**:271–287 DOI 10.1002/1097-0134(20001115)41:3<271::AID-PROT10>3.0.CO;2-Z.

DeLano WL. 2002. The PyMOL molecular graphics system. Version 1.5.0.4. New York: Schrödinger, LLC. *Available at http://www.pymol.org*.

Dixit A, Verkhivker GM. 2011. Computational modeling of allosteric communication reveals organizing principles of mutation-induced signaling in ABL and EGFR Kinases. *PLOS Computational Biology* **7(10)**:e1002179 DOI 10.1371/journal.pcbi.1002179.

Dong Q-W, Wang X-L, Lin L. 2007. Methods for optimizing the structure alphabet sequences of proteins. *Computers in Biology and Medicine* **37(11)**:1610–1616 DOI 10.1016/j.compbiomed.2007.03.002.

Dudev M, Lim C. 2007. Discovering structural motifs using a structural alphabet: application to magnesium-binding sites. *BMC Bioinformatics* **8(1)**:106 DOI 10.1186/1471-2105-8-106.

Etchebest C, Benros C, Hazout S, De Brevern AG. 2005. A structural alphabet for local protein structures: improved prediction methods. *Proteins: Structure, Function, and Bioinformatics* **59(4)**:810–827 DOI 10.1002/prot.20458.

Faure G, Bornot A, De Brevern AG. 2008. Protein contacts, inter-residue interactions and side-chain modelling. *Biochimie* **90(4)**:626–639 DOI 10.1016/j.biochi.2007.11.007.

Frauenfelder H, Sligar S, Wolynes P. 1991. The energy landscapes and motions of proteins. *Science* **254(5038)**:1598–1603 DOI 10.1126/science.1749933.

Gelly J-C, Joseph AP, Srinivasan N, De Brevern AG. 2011. iPBA: a tool for protein structure comparison using sequence alignment strategies. *Nucleic Acids Research* **39(suppl)**:W18–W23 DOI 10.1093/nar/gkr333.

Ghosh A, Vishveshwara S. 2007. A study of communication pathways in methionyl-tRNA synthetase by molecular dynamics simulations and structure network analysis. *Proceedings of the National Academy of Sciences of the United States of America* **104(40)**:15711–15716 DOI 10.1073/pnas.0704459104.

Ghouzam Y, Postic G, De Brevern AG, Gelly J-C. 2015. Improving protein fold recognition with hybrid profiles combining sequence and structure evolution. *Bioinformatics* **31(23)**:3782–3789 DOI 10.1093/bioinformatics/btv462.

Ghouzam Y, Postic G, Guerin P-E, De Brevern AG, Gelly J-C. 2016. ORION: a web server for protein fold recognition and structure prediction using evolutionary hybrid profiles. *Scientific Reports* **6(1)**:Article 28268 DOI 10.1038/srep28268.

Goguet M, Narwani TJ, Peterman R, Jallu V, De Brevern AG. 2017. *In silico* analysis of Glanzmann variants of Calf-1 domain of alphaIIb/beta3 integrin revealed dynamic allosteric effect. *Scientific Reports* **7(1)**:Article 8001 DOI 10.1038/s41598-017-08408-w.

Gowers RJ, Linke M, Barnoud J, Reddy TJE, Melo MN, Seyler SL, Domaski J, Dotson DL, Buchoux S, Kenney IM, Beckstein O. 2016. MDAnalysis: a python package for the rapid analysis of molecular dynamics simulations. In: Sebastian B, Scott R, eds. *Proceedings of the 15th Python in Science Conference*. 98–105.

Holscher E, Leifer C, Grace B. 2010. Read the Docs. *Available at https://readthedocs.org/*.

Jallu V, Bertrand G, Bianchi F, Chenet C, Poulain P, Kaplan C. 2013. The αIIb p.Leu841Met (Cab3a+) polymorphism results in a new human platelet alloantigen involved in neonatal alloimmune thrombocytopenia. *Transfusion* **53(3)**:554–563 DOI 10.1111/j.1537-2995.2012.03762.x.

Jallu V, Poulain P, Fuchs PFJ, Kaplan C, De Brevern AG. 2012. Modeling and molecular dynamics of HPA-1a and -1b polymorphisms: effects on the structure of the *B3* subunit of the αIIbβ3 Integrin. *PLOS ONE* **7(11)**:e47304 DOI 10.1371/journal.pone.0047304.

Jallu V, Poulain P, Fuchs PFJ, Kaplan C, De Brevern AG. 2014. Modeling and molecular dynamics simulations of the V33 variant of the integrin subunit *B3*: structural Comparison with the L33 (HPA-1a) and P33 (HPA-1b) variants. *Biochimie* **105**:84–90 DOI 10.1016/j.biochi.2014.06.017.

Joseph AP, Agarwal G, Mahajan S, Gelly J-C, Swapna LS, Offmann B, Cadet F, Bornot A, Tyagi M, Valadié H, Schneider B, Etchebest C, Srinivasan N, De Brevern AG. 2010. A short survey on protein blocks. *Biophysical Reviews* **2(3)**:137–147 DOI 10.1007/s12551-010-0036-1.

Joseph AP, Srinivasan N, De Brevern AG. 2011. Improvement of protein structure comparison using a structural alphabet. *Biochimie* **93(9)**:1434–1445 DOI 10.1016/j.biochi.2011.04.010.

Joseph AP, Srinivasan N, De Brevern AG. 2012. Progressive structure-based alignment of homologous proteins: adopting sequence comparison strategies. *Biochimie* **94(9)**:2025–2034 DOI 10.1016/j.biochi.2012.05.028.

Kaplan C. 2006. Neonatal alloimmune thrombocytopenia. In: McCrae KR, ed. *Thrombocytopenia*. Milton Park: Taylor & Francis Group, 223–244.

Kaplan C, Freedman J. 2007. Platelets. In: Michelson AD, ed. *Platelets*. London: Academic Press, 971–984.

Laine E, Auclair C, Tchertanov L. 2012. Allosteric communication across the native and mutated kit receptor tyrosine kinase. *PLOS Computational Biology* **8(8)**:e1002661 DOI 10.1371/journal.pcbi.1002661.

Léonard S, Joseph AP, Srinivasan N, Gelly J-C, De Brevern AG. 2014. mulPBA: an efficient multiple protein structure alignment method based on a structural alphabet. *Journal of Biomolecular Structure and Dynamics* **32(4)**:661–668 DOI 10.1080/07391102.2013.787026.

Li Q, Zhou C, Liu H. 2009. Fragment-based local statistical potentials derived by combining an alphabet of protein local structures with secondary structures and solvent accessibilities. *Proteins: Structure, Function, and Bioinformatics* **74(4)**:820–836 DOI 10.1002/prot.22191.

Lindahl E, Hess B, Van der Spoel D. 2001. GROMACS 3.0: a package for molecular simulation and trajectory analysis. *Journal of Molecular Modeling* **7(8)**:306–317 DOI 10.1007/s008940100045.

Lubienski MJ, Bycroft M, Freund SM, Fersht AR. 1994. Three-dimensional solution structure and 13C assignments of barstar using nuclear magnetic resonance spectroscopy. *Biochemistry* **33(30)**:8866–8877.

Michaud-Agrawal N, Denning EJ, Woolf TB, Beckstein O. 2011. MDAnalysis: a toolkit for the analysis of molecular dynamics simulations. *Journal of Computational Chemistry* **32(10)**:2319–2327 DOI 10.1002/jcc.21787.

Nguyen LAT, Dang XT, Le TKT, Saethang T, Tran VA, Ngo DL, Gavrilov S, Nguyen NG, Kubo M, Yamada Y, Satou K. 2014. Predicting *B*eta-Turns and *B*eta-turn types using a novel over-sampling approach. *Journal of Biomedical Science and Engineering* **07(11)**:927–940 DOI 10.4236/jbise.2014.711090.

Offmann B, Tyagi M, De Brevern A. 2007. Local protein structures. *Current Bioinformatics* **2(3)**:165–202 DOI 10.2174/157489307781662105.

Pandini A, Fornili A, Fraternali F, Kleinjung J. 2013. GSATools: analysis of allosteric communication and functional local motions using a structural alphabet. *Bioinformatics* **29(16)**:2053–2055 DOI 10.1093/bioinformatics/btt326.

Poulain P, De Brevern AG. 2012. Model of the Beta3 Subunit of Integrin alphaIIb/Beta3. *Available at https://dx.doi.org/10.6084/m9.figshare.104602.v2*.

Python Software Foundation. 2010. Python Language Reference, Version 2.7. *Available at http://www.python.org*.

Rangwala H, Kauffman C, Karypis G. 2009. svmPRAT: SVM-based protein residue annotation toolkit. *BMC Bioinformatics* **10(1)**:439 DOI 10.1186/1471-2105-10-439.

Sevcík J, Urbanikova L, Dauter Z, Wilson KS. 1998. Recognition of RNase Sa by the inhibitor barstar: structure of the complex at 1.7 A resolution. *Acta Crystallographica. Section D, Biological Crystallography* **54(Pt 5)**:954–963.

Suresh V, Ganesan K, Parthasarathy K. 2013. A protein block based fold recognition method for the annotation of twilight zone sequences. *Protein and Peptide Letters* **20(3)**:249–254.

Suresh V, Parthasarathy S. 2014. SVM-PB-Pred: SVM based protein block prediction method using sequence profiles and secondary structures. *Protein & Peptide Letters* **21(8)**:736–742 DOI 10.2174/0929866511320999064.

Thomas A, Deshayes S, Decaffmeyer M, Van Eyck MH, Charloteaux B, Brasseur R. 2006. Prediction of peptide structure: how far are we? *Proteins: Structure, Function, and Bioinformatics* **65(4)**:889–897 DOI 10.1002/prot.21151.

Van der Spoel D, Lindahl E, Hess B, Groenhof G, Mark AE, Berendsen HJC. 2005. GROMACS: fast, flexible, and free. *J Comput Chem* **26(16)**:1701–1718 DOI 10.1002/jcc.20291.

Van Rossum G. 1995. Python Tutorial. Technical Report CS-R9526. Centrum voor Wiskunde en Informatica (CWI), Amsterdam. *Available at https://ir.cwi.nl/pub/5007/05007D.pdf* .

Zhu J, Luo B-H, Xiao T, Zhang C, Nishida N, Springer TA. 2008. Structure of a complete integrin ectodomain in a physiologic resting state and activation and deactivation by applied forces. *Molecular Cell* **32(6)**:849–861 DOI 10.1016/j.molcel.2008.11.018.

Zimmermann O, Hansmann UHE. 2008. LOCUSTRA: accurate prediction of local protein structure using a two-layer support vector machine approach. *Journal of Chemical Information and Modeling* **48(9)**:1903–1908 DOI 10.1021/ci800178a.